PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:

Jean-Marie GATTO
Thierry BRUNET DE COURSSOU
Pierre-Jean BENEY

Serial No.: 10/656,631

Filed: September 4, 2003

For: **UNIVERSAL GAME SERVER**

Atty. Docket No.: CYBS5872

Examiner: Meagan J. THOMASSON

Art Unit: 3714

Confirmation No.: 8128

Customer No.: 22430

PROPOSED AMENDMENT IN PREPARATION OF TELEPHONE INTERVIEW OF 2/1/2011
- **Do Not Enter**

Mail Stop Amendment
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Responsive to the Office Action mailed October 13, 2010, please amend the above application as indicated below.

A complete listing of the **claims** begins on page **2** of this paper.

Remarks begin on page **25** of this paper.

IN THE CLAIMS:

1. **(Currently Amended)** An online gaming system, comprising:

at least two central servers, each of the at least two servers being coupled to a communication network, and

at least one gaming machine coupled to the at least two central servers through the communication network in a client-server configuration in which each of the at least one gaming machine is a client to the at least two central servers, each of the at least one gaming machine being configured to play at least one game and to carry out a game transaction for each game played and to commit each game transaction to each of the at least two central servers by sending ~~a a single~~ **a single** transaction packet to each of the at least two central servers, each single transaction packet sent to each of the at least two central servers including an identical inbound game payload wherein each of the at least two central servers, upon receipt of the inbound game payload, are configured to return a single outbound game payload to the gaming machine having sent the transaction packet, the outbound game payload enabling the gaming machine having sent the transaction packet to complete the game transaction and wherein the at least one gaming machine is configured such that a first arriving outbound payload received by the at least one gaming machine is effective to complete the game transaction, irrespective of when and if a second later arriving outbound payload is received by the at least one gaming machine.

2. **(Original)** The online gaming system according to claim 1, wherein each of the at least two central servers returns a game transaction commit acknowledgment to the at least one gaming machine.

3. **(Previously Presented)** The online gaming system according to claim 2, wherein the at least one gaming machine is configured to acknowledge to a player a validity of the game

transaction upon receipt of at least one game transaction commit acknowledgment during a predetermined timeout period following the commit of the game transaction to each of the at least two central servers.

4. **(Previously Presented)** The online gaming system according to claim 1, wherein the inbound game payload includes at least one of a gaming machine ID, a user/player ID, a transaction GUID, a gaming machine originating/return address, a game ID, a game bet, and an amount wagered.

5. **(Original)** The online gaming system according to claim 1, wherein the at least one gaming machine is configured to be an active participant in a fault tolerance of the online gaming system.

6. **(Previously Presented)** The online gaming system according to claim 1, wherein the at least one gaming machine is configured to record a synchronization log that includes identifiers of any transactions that were not acknowledged by a non-responding one of the at least two central servers after a predetermined timeout, the synchronization log being used to subsequently send the unacknowledged transactions to the non-responding one of the at least two central servers.

7. **(Previously Presented)** The online gaming system according to claim 6, wherein the non-responding one of the at least two central servers is configured to be synchronized by receiving the unacknowledged transactions directly from the at least one gaming machine subsequent to communication being re-established therewith.

8. **(Original)** The online gaming system according to claim 1, wherein the communication network is the Internet and wherein a protocol to transport a payload of each game transaction is UDP.

9. **(Original)** The online gaming system according to claim 1, wherein the at least two central servers and the at least one gaming machine are configured to support instant-draw and deferred-draw of random events.

10. **(Original)** The online gaming system according to claim 1, wherein the at least two central servers are geographically remote from one another.

11. **(Previously Presented)** The online gaming system according to claim 1, wherein each of the at least two central servers comprises a trusted transactional cache, the trusted transactional cache being configured to process each committed game transaction received directly and independently from each of the at least one gaming machine, and to provide real time persistent storage and logging of aspects of each committed game transaction.

12. **(Original)** The online gaming system according to claim 1, wherein the at least two central servers further comprise at least one of a trusted transactional cache, a business server and a logistic support server.

13. **(Withdrawn)** An online gaming system, comprising:

a communication network;

at least two geographically dispersed central servers, each of the at least two geographically dispersed central servers being coupled to the communication network,

at least two gaming machines, each of the at least two gaming machines being coupled to the communication network and being configured to carry out a game transaction for each game played, the at least two gaming machines being configured to carry out load balancing when committing the game transactions to the at least two geographically dispersed central servers over the communication network, the load balancing including each gaming machine selecting only one of the at least two geographically dispersed central servers to which to commit the game transaction.

14. **(Canceled)**

15. **(Withdrawn)** The online gaming system according to claim 13, wherein the communication network is the Internet and wherein a protocol to transport a payload of each game transaction is UDP.

16. **(Withdrawn)** The online gaming system according to claim 13, wherein the at least two central servers and the at least two gaming machines are configured to support instant-draw and deferred-draw of random events.

17. **(Withdrawn)** The online gaming system according to claim 13, wherein each of the at least two geographically dispersed central servers each further comprise a trusted transactional cache, the trusted transactional cache being configured to process each committed game transaction received directly and independently from each of the at least one gaming machine, and to provide real time persistent storage and logging of aspects of each committed game transaction.

18. **(Withdrawn)** The online gaming system according to claim 13, wherein the at least two geographically dispersed central servers each further comprise at least one of a trusted transactional cache, a business server and a logistic support server.

19. **(Withdrawn)** An online gaming system, comprising:

a communication network;

a plurality of gaming machines, each of the plurality of gaming machines being configured to carry out game transactions and being coupled to the communication network, and

N geographically dispersed central servers, wherein N is equal to at least two, each of the N geographically dispersed central servers being coupled to the communication network, selected ones of the plurality of gaming machines being further configured to perform load balancing when committing transactions to the N geographically dispersed central servers, the load balancing including having each gaming machine select at least one of the N geographically dispersed central servers to which to commit the game transactions.

20. **(Canceled)**

21. **(Withdrawn)** The online gaming system according to claim 19, wherein each of the N geographically dispersed central servers is configured to return a game transaction commit acknowledgment to the gaming machine that initiated the transaction commit over the communication network.

22. **(Withdrawn)** The online gaming system according to claim 21, wherein the gaming machine acknowledges to the player the validity of the game transaction upon receipt of at least one game transaction commit acknowledgment during a predetermined timeout period following the commit of the game transaction to each of the N geographically dispersed central servers.

23. **(Withdrawn)** The online gaming system according to claim 19, wherein each game transaction committed to each of the N geographically dispersed central servers have an identical inbound game payload comprising at least a selected set of the at least one gaming

machine ID, the user/player ID, the transaction GUID, the gaming machine originating/return address, the game ID, the game bet, and the amount wagered.

24. **(Withdrawn)** The online gaming system according to claim 19, wherein the communication network includes the Internet and wherein a protocol to transport a payload of each of the game transactions is UDP.

25. **(Withdrawn)** The online gaming system according to claim 19, wherein the N geographically dispersed central servers and the plurality of gaming machines are configured to support instant-draw and deferred-draw of random events.

26. **(Withdrawn)** The online gaming system according to claim 19, wherein each of the N geographically dispersed central servers each further comprises a trusted transactional cache, the trusted transactional cache being configured to process each committed game transaction, and to provide real time, secure and persistent storage and logging of aspects of each committed game transaction.

27. **(Withdrawn)** The online gaming system according to claim 19, each of the N geographically dispersed central servers further comprise at least one of a trusted transactional cache, a business server and a logistic support server.

28. **(Withdrawn)** An online gaming system, comprising:

a plurality of gaming machines, each of the plurality of gaming machines being configured to generate and send an inbound transaction packet that includes an inbound transaction payload across at least one of a plurality of communication networks according to one of a plurality of communication protocols;

at least one central server coupled to the plurality of communication networks and to each of the at least one central servers, the at least one central server including:

at least one transaction engine configured to process inbound transaction payloads to generate corresponding outbound transaction payloads;

a personality front end, the personality front end being configured to interface with each of the plurality of communication networks to receive inbound transaction packets from the plurality of gaming machines, to extract the inbound transaction payloads from the received inbound transaction packets, to submit the extracted inbound payloads to the at least one transaction engine, to generate outbound transaction packets that include the corresponding outbound transaction payloads and to send the generated outbound transaction packets to a selected one of the plurality of gaming machines.

29. **(Withdrawn)** The online gaming system according to claim 28, wherein the inbound transaction payload comprises at least one of a gaming machine ID, a user/player ID, a transaction GUID, a terminal originating/return address, a game ID, a game bet, and an amount wagered.

30. **(Withdrawn)** The online gaming system according to claim 28, wherein the personality front end is further configured to transcode specific transaction payloads produced by the plurality of gaming terminals into generic transaction payloads.

31. **(Withdrawn)** The online gaming system according to claim 28, wherein the plurality of communication networks include at least one of dial-up, X25, Frame Relay, leased line, Internet and VPN.

32. **(Withdrawn)** The online gaming system according to claim 28, wherein said one of the plurality of communication protocols is selected from one of proprietary, X25, TCP/IP, UDP, HTTP, XML and SOAP protocols.

33. **(Withdrawn)** A game random number generator for supplying random game numbers to a gaming machine, comprising:

at least one hardware number generator configured to provide random number seeds at a predetermined rate, and

at least one pseudo-random number generator coupled to the at least one hardware number generator, the at least one pseudo-random number generator being configured to generate the random game numbers from the random number seeds generated by the at least one hardware number generator.

34. **(Withdrawn)** The game random number generator according to claim 33, further comprising a first trusted log configured to securely log all of random number seeds generated by the at least one hardware number generator.

35. **(Withdrawn)** The game random number generator according to claim 33, further comprising a second trusted log configured to securely log all of random game numbers generated by the at least one pseudo-random number generator.

36. **(Withdrawn)** The game random number generator according to claim 33, wherein the at least one pseudo-random number generator is configured to supply game random numbers on demand for each individual game draw within the gaming machine.

37. **(Withdrawn)** The game random number generator according to claim 33, further comprising at least one game result assembler coupled to the at least one pseudo-random number

generator, the at least one game result assembler being configured to receive random game numbers produced by the at least one pseudo-random number generator and to generate ranging random game numbers.

38. **(Withdrawn)** The game random number generator according to claim 33, wherein the at least one hardware random number generator is one of:

a RNG of Intel 8XX series of PC motherboard chipsets, the chipset being integrated on a motherboard of a computer within the gaming machine;

a RNG of a secure smart card communicating with the computer within the gaming machine;

a RNG of a secure smart device communicating with the computer of the gaming machine;

a RNG of a processor compliant with Microsoft Next-Generation Secure Computing Base, the processor being integrated on the motherboard of the computer of the gaming machine;

a RNG of a motherboard chipset compliant with Microsoft Next-Generation Secure Computing Base, the chipset being integrated on the motherboard of the computer of the gaming machine;

a RNG of a security plug-in device communicating with the computer within the gaming machine, and

a RNG of an add-on card or add-on board security device communicating with the computer within the gaming machine.

39. **(Withdrawn)** A gaming system comprising:

at least one gaming machine;

at least one central game server coupled to the at least one gaming machine over a network, the at least one central game server including:

at least one hardware number generator configured to provide random number seeds at a predetermined rate, and

at least one pseudo-random number generator coupled to the at least one hardware number generator, the at least one pseudo-random number generator being configured to generate, on demand, the random game numbers from the random number seeds generated by the at least one hardware number generator.

40. **(Withdrawn)** The gaming system according to claim 39, further comprising a first trusted log configured to securely log all of random number seeds generated by the at least one hardware number generator.

41. **(Withdrawn)** The gaming system according to claim 39, further comprising a second trusted log configured to securely log all of random game numbers generated by the at least one pseudo-random number generator.

42. **(Withdrawn)** The gaming system according to claim 39, wherein the at least one pseudo-random number generator is configured to supply game random numbers on demand for each individual game draw within the gaming machine.

43. **(Withdrawn)** The gaming system according to claim 39, further comprising at least one game result assembler coupled to the at least one pseudo-random number generator, the at least one game result assembler being configured to receive random game numbers produced by the at least one pseudo-random number generator and to generate ranging random game numbers.

44. **(Withdrawn)** The gaming system according to claim 39, wherein the at least one hardware random number generator is one of:

a RNG of Intel 8XX series of PC motherboard chipsets, the chipset being integrated on a motherboard of a computer within the gaming machine;

a RNG of a secure smart card communicating with the computer within the gaming machine;

a RNG of a secure smart device communicating with the computer of the gaming machine;

a RNG of a processor compliant with Microsoft Next-Generation Secure Computing Base, the processor being integrated on the motherboard of the computer of the gaming machine;

a RNG of a motherboard chipset compliant with Microsoft Next-Generation Secure Computing Base, the chipset being integrated on the motherboard of the computer of the gaming machine;

a RNG of a security plug-in device communicating with the computer within the gaming machine, and

a RNG of an add-on card or add-on board security device communicating with the computer within the gaming machine.

45. **(Withdrawn)** A gaming system comprising

at least one gaming machine, including:

at least one first hardware number generator configured to provide random number seeds at a predetermined rate, and

at least one first pseudo-random number generator coupled to the at least one first hardware number generator, the at least one first pseudo-random number generator being configured to generate, on demand, the random game numbers from the random number seeds generated by the at least one first hardware number generator for each game draw performed at the at least one gaming machine;

at least one central game server coupled to the at least one gaming machine, the central game server including:

at least one second hardware number generator configured to provide random number seeds at a predetermined rate, and

at least one second pseudo-random number generator coupled to the at least one second hardware number generator, the at least one second pseudo-random number generator being configured to generate, on demand, the random game numbers from the random number seeds generated by the at least one second hardware number generator for each game draw performed at the at least one gaming machine.

46. **(Withdrawn)** The gaming system according to claim 45, further comprising:

a first trusted log configured to securely log all of random number seeds generated by the at least one first hardware number generator, and

a second trusted log configured to securely log all of random number seeds generated by the at least one second hardware number generator.

47. **(Withdrawn)** The gaming system according to claim 45, further comprising:

a third trusted log configured to securely log all of random game numbers generated by the at least one first pseudo-random number generator, and

a fourth trusted log configured to securely log all of random game numbers generated by the at least one second pseudo-random number generator.

48. **(Withdrawn)** The gaming system according to claim 45, wherein first and second hardware random number generators are identical.

49. **(Withdrawn)** The gaming system according to claim 45, wherein first and second pseudo random number generators are identical.

50. **(Withdrawn)** The gaming system according to claim 45, wherein that at least one gaming machine is configured to select at least one random game number for each game draw from the at least one first pseudo-random number generator or from the second pseudo-random number generator.

51. **(Withdrawn)** A gaming system according to claim 45, further comprising

at least one game result assembler coupled to the at least one first pseudo-random number generator or to the at least one second pseudo-random number generator, the at least one game result assembler being configured to receive random game numbers produced by the first or second pseudo-random number generators and to generate ranging random game numbers.

52. **(Withdrawn)** The gaming system according to claim 45, wherein the first or second hardware random number generator is one of:

a RNG of Intel 8XX series of PC motherboard chipsets, the chipset being integrated on a motherboard of a computer within the gaming machine;

a RNG of a secure smart card communicating with the computer within the gaming machine;

a RNG of a secure smart device communicating with the computer of the gaming machine;

a RNG of a processor compliant with Microsoft Next-Generation Secure Computing Base, the processor being integrated on the motherboard of the computer of the gaming machine;

a RNG of a motherboard chipset compliant with Microsoft Next-Generation Secure Computing Base, the chipset being integrated on the motherboard of the computer of the gaming machine;

a RNG of a security plug-in device communicating with the computer within the gaming machine, and

a RNG of an add-on card or add-on board security device communicating with the computer within the gaming machine.

53. **(Withdrawn)** A gaming machine configured to execute game draws whose outcome depend upon random game numbers, the gaming machine comprising:

at least one hardware number generator configured to provide random number seeds at a predetermined rate, and

at least one pseudo-random number generator coupled to the at least one hardware number generator, the at least one pseudo-random number generator being configured to generate the random game numbers from the random number seeds generated by the at least one hardware number generator.

54. **(Withdrawn)** The gaming machine according to claim 53, further comprising a first trusted log configured to securely log all of random number seeds generated by the at least one hardware number generator.

55. **(Withdrawn)** The gaming machine according to claim 53, further comprising a second trusted log configured to securely log all of random game numbers generated by the at least one pseudo-random number generator.

56. **(Withdrawn)** A gaming system comprising:

a communication network;

at least one central web server, each of the at least one central web server being coupled to the network,

at least one central transaction server, each of the at least one central transaction server being coupled to the network and,

at least one web browser based gaming machine coupled to the communication network, each of the at least one web browser based gaming machine comprising:

a standard web browser being configured to display rich page content and animations of the games produced by the at least one central web server, and

a plug-in for the standard web browser, the plug-in being configured to carry out a game transaction for each game played and to commit each game transaction to the at least one central transaction server.

57. **(Withdrawn)** A gaming system according to claim 56 wherein the communication network includes the Internet.

58. **(Withdrawn)** A gaming system according to claim 56, wherein the plug-in is configured to complete the game transaction upon receipt of a validation transaction from the at least one central transaction server.

59. **(Withdrawn)** A gaming system according to claim 56, wherein the committed game transaction includes an inbound game payload comprising at least one of a gaming machine ID, a user/player ID, a transaction GUID, a gaming machine originating/return address, a game ID, a game bet, and an amount wagered.

60. **(Withdrawn)** A gaming system according to claim 59 whereby the validation transaction from the at least one central transaction server includes an outbound packet

comprising at least one of a gaming machine ID, a user/player ID, a transaction GUID, and an outcome of the game.

61. **(Withdrawn)** A gaming system according to claim 56, wherein the plug-in is further configured to commit each game transaction to each of the at least one central transaction servers.

62. **(Canceled)**

63. **(Previously Presented)** The online gaming system of claim 1, wherein the at least one gaming machine is configured to initiate and terminate the game transaction.

64. **(Previously Presented)** The online gaming system of claim 1 wherein, as between the at least one gaming machine and the at least two central servers, the at least one gaming machine is configured as sole master of the game transaction.

65. **(Previously Presented)** The online gaming system of claim 1 wherein, as between the at least one gaming machine and the at least two central servers, only the at least one gaming machine is configured for recovery from network communications errors occurring during the game transaction.

66. **(Withdrawn)** The online gaming system of claim 13, wherein the at least one gaming machine is configured to initiate and terminate the game transaction.

67. **(Withdrawn)** The online gaming system of claim 13 wherein, as between the at least two geographically dispersed central servers and the at least two gaming machines, the at least two gaming machines are configured as masters of the game transactions.

68. **(Withdrawn)** The online gaming system of claim 13 wherein, as between the at least two geographically dispersed central servers and the at least two gaming machines, only the

at least two gaming machine are configured for recovery from network communication errors occurring during the game transactions.

69. **(Withdrawn)** The online gaming system of claim 19, wherein each of the plurality of gaming machines is configured to initiate and terminate the game transactions.

70. **(Withdrawn)** The online gaming system of claim 19 wherein, as between the plurality of gaming machines and the N geographically dispersed central servers, the plurality of gaming machines are configured as sole masters of game transactions.

71. **(Withdrawn)** The online gaming system of claim 19, wherein, as between the plurality of gaming machines and the N geographically dispersed central servers, only the plurality of gaming machines are configured for recovery from network communication errors occurring during game transactions.

72. **(Withdrawn)** The online gaming system of claim 17, wherein each of the at least two gaming machines is configured to record a synchronization log that includes identifiers of any transactions that were not acknowledged by a non-responding trusted transactional cache after a predetermined timeout, the synchronization log being used to subsequently send the unacknowledged transactions to the non-responding trusted transactional cache.

73. **(Withdrawn)** The online gaming system according to claim 72, wherein the non-responding trusted transactional cache is further configured to be synchronized by receiving the unacknowledged transactions directly from the at least one gaming machine subsequent to communication being re-established therewith.

74. **(Withdrawn)** The online gaming system according to claim 72, wherein each trusted transactional cache includes a synchronization engine and wherein the non-responding

trusted transactional cache is further configured to be synchronized by receiving the unacknowledged transactions directly from the synchronization engine of a responding trusted transactional cache.

75. **(Withdrawn)** The online gaming system of claim 26, wherein each of the plurality of gaming machines is configured to construct a synchronization log that includes identifiers of any transactions that were not acknowledged by a non-responding trusted transactional cache after a predetermined timeout, the synchronization log being used to subsequently send the unacknowledged transactions to the non-responding trusted transactional cache.

76. **(Withdrawn)** The online gaming system according to claim 75, wherein the non-responding trusted transactional cache is further configured to be synchronized by receiving the unacknowledged transactions directly from the at least one gaming machine subsequent to communication being re-established therewith.

77. **(Withdrawn)** The online gaming system according to claim 75, wherein each trusted transactional cache includes a synchronization engine and wherein the non-responding trusted transactional cache is further configured to be synchronized by receiving the unacknowledged transactions directly from the trusted transactional cache of a responding trusted transactional cache.

78. **(Previously Presented)** The online gaming system according to claim 1, wherein each of the at least two central servers includes a synchronization engine and wherein the non-responding one of the at least two central servers is configured to be synchronized by receiving the unacknowledged transactions directly from the synchronization engine of a responding one of the at least two central servers.

79. **(Currently Amended)** An online gaming system, comprising:

~~**a communication network;**~~

at least two central servers, each of the at least two servers being coupled to ~~**the**~~ **a communication** network, each of the at least two central servers including a synchronization engine and

at least one gaming machine coupled to the communication network, each of the at least one gaming machine being configured to play at least one game and to carry out a game transaction for each game played and to commit each game transaction to each of the at least two central servers by sending a single transaction packet to each of the at least two central servers, each single transaction packet sent to each of the at least two central servers including an identical inbound game payload, wherein each of the two central servers are configured such that any transaction packet that is not acknowledged by a non-responding one of the at least two central servers is sent directly from the synchronization engine of a responding one of the at least two central servers to the synchronization engine of the non-responding central server.

80. **(Previously Presented)** The online gaming system according to claim 79, wherein each of the at least two central servers returns a game transaction commit acknowledgment to the at least one gaming machine.

81. **(Previously Presented)** The online gaming system according to claim 80, wherein the at least one gaming machine is configured to acknowledge to a player a validity of the game transaction upon receipt of a first arriving game transaction commit acknowledgment during a predetermined timeout period following the commit of the game transaction to each of the at least two central servers.

82. **(Previously Presented)** The online gaming system according to claim 79, wherein the game payload includes at least one of a gaming machine ID, a user/player ID, a transaction GUID, a gaming machine originating/return address, a game ID, a game bet, and an amount wagered.

83. **(Previously Presented)** The online gaming system according to claim 79, wherein the at least one gaming machine is configured to be an active participant in a fault tolerance of the online gaming system.

84. **(Previously Presented)** The online gaming system according to claim 79, wherein the non-responding one of the at least two central servers is configured to be synchronized by receiving the unacknowledged transactions directly from the at least one gaming machine subsequent to communication being re-established therewith.

85. **(Previously Presented)** The online gaming system according to claim 79, wherein the communication network is the Internet and wherein a protocol to transport a payload of each game transaction is UDP.

86. **(Previously Presented)** The online gaming system according to claim 79, wherein the at least two central servers and the at least one gaming machine are configured to support instant-draw and deferred-draw of random events.

87. **(Previously Presented)** The online gaming system according to claim 79, wherein the at least two central servers are geographically remote from one another.

88. **(Previously Presented)** The online gaming system according to claim 79, wherein each of the at least two central servers comprises a trusted transactional cache, the trusted transactional cache being configured to process each committed game transaction

received directly and independently from each of the at least one gaming machine, and to provide real time persistent storage and logging of aspects of each committed game transaction.

89. **(Previously Presented)** The online gaming system according to claim 79, wherein the at least two central servers further comprise at least one of a trusted transactional cache, a business server and a logistic support server.

90. **(Previously Presented)** The online gaming system of claim 79, wherein the at least one gaming machine is configured to initiate and terminate the game transaction.

91. **(Previously Presented)** The online gaming system of claim 79, wherein, as between the at least one gaming machine and the at least two central servers, the at least one gaming machine is configured as sole master of the game transaction.

92. **(Previously Presented)** The online gaming system of claim 79, wherein, as between the at least one gaming machine and the at least two central servers, only the at least one gaming machine is configured for recovery from network communications errors occurring during the game transaction.

93-107. **(Canceled)**

108. **(Previously Presented)** A computer-implemented method of carrying out a game transaction, comprising the steps of:

providing at least two central servers;

providing a gaming machine configured to enable a game to be played;

coupling the gaming machine and each of the at least two provided central servers to a communication network in a client-server configuration in which the gaming machine is a client to the at least two central servers;

carrying out, in the gaming machine, a game transaction for each game played;

committing each game transaction to each of the at least two central servers by sending a single transaction packet to each of the at least two central servers, each single transaction packet sent to each of the at least two central servers including an identical inbound game payload;

returning, by each of the at least two central servers, a single outbound game payload to the gaming machine upon receipt of the inbound game payload, and

completing the game transaction, by the gaming machine, upon receipt of a first in time received outbound game payload from one of the at least two central server, irrespective of whether and when a later in time outbound game payload is received from another one or ones of the at least two central servers.

109. **(Previously Presented)** A computer-implemented method of carrying out a game transaction, comprising the steps of:

providing at least two central servers;

providing a gaming machine configured to enable a game to be played;

coupling the gaming machine and each of the at least two provided central servers to a communication network

carrying out a game transaction for each game played;

committing each game transaction to each of the at least two central servers by sending a single transaction packet to each of the at least two central servers, each single transaction packet sent to each of the at least two central servers including an identical inbound game payload;

completing the game transaction, by the gaming machine, upon receipt of a first in time received outbound game payload from one of the at least two central server, irrespective of whether and when a later in time outbound game payload is received from another one or ones of the at least two central servers, and

recording, in the gaming machine, a synchronization log that includes identifiers of any transactions that were not acknowledged by a non-responding one of the at least two central servers after a predetermined timeout, the synchronization log being configured to enable the gaming machine to subsequently send the unacknowledged transactions to the non-responding one of the at least two central servers.

110. **(Canceled)**

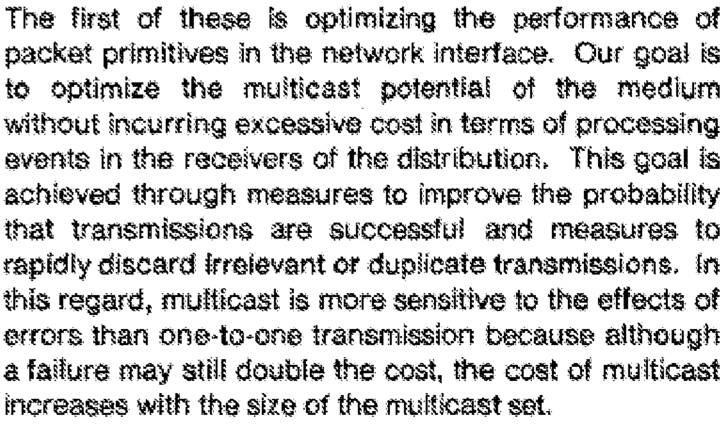
REMARKS

This (proposed) Amendment is in response to the Final Office Action mailed October 13, 2010 and is submitted in advance of the telephone interview of Feb. 1, 2011. Please do not enter.

Claims 1-5, 8-12, 63-65 and 108 were rejected as being unpatentable over Mockapetris in view of Nguyen. Reconsideration and withdrawal of these rejections are respectfully requested.

With regard to Mockapetris' statement on page 152, Column 2 that an important goal of his system is to "optimize the multicast potential of the medium without incurring excessive cost in terms of processing events in the receivers of the distribution" and the Examiner assertion that "With that goal explicit, a single acknowledgment would be sufficient to complete a transaction", the undersigned respectfully submits the following.

Mockapetris seeks to optimize the multi-casting of the messages without incurring excessive costs in terms of processing events in the receivers of the distribution, by improving the probability that the transmissions are successful and discarding duplicate **transmissions** (right-hand col., page 152):



> The first of these is optimizing the performance of packet primitives in the network interface. Our goal is to optimize the multicast potential of the medium without incurring excessive cost in terms of processing events in the receivers of the distribution. This goal is achieved through measures to improve the probability that transmissions are successful and measures to rapidly discard irrelevant or duplicate transmissions. In this regard, multicast is more sensitive to the effects of errors than one-to-one transmission because although a failure may still double the cost, the cost of multicast increases with the size of the multicast set.

Therefore, Mockapetris seeks to 1) optimize the multi-cast transmission of the messages (in a manner that does not impose excessive burdens on the receivers thereof) and 2) to do so by ("This goal is achieved through...") a) improving the probability that the transmissions are successful and b) discarding irrelevant or duplicate transmissions. These, then are the outgoing transmissions of the multicast messages.

These goals of maximizing the **sending of the messages** are discussed in the "Packet primitives strategies and costs" section of Mockapetris:

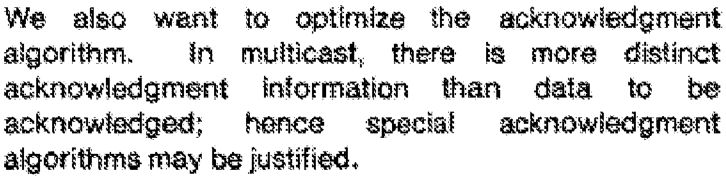
> **Packet primitives strategies and costs Several techniques for improving interface performance are already in use in various systems. Interfaces should recognize multicast addresses instead of a single broadcast address; thus hosts that are not in the multicast set won't have to expend time to discard extraneous packets. Network interfaces can minimize packet loss through full duplex operation and by buffering strategies that allow reception of back-to-back packets from the medium. A fairly simple extension to this scheme would be to reserve a buffer for each multicast connection so that multicast distributions would never be discarded due to lack of resources.**
>
> **In situations such as saturation, where the ACK reliability isn't a problem, but the distribution may still need retransmission, duplicate detection can be made automatic by using two multicast addresses and a variation on the alternating bit protocol[12]. In this scheme, called *parity,* the sender uses one address for the initial transmission *and all retransmissions* of a message, and then switches to the other address for the next message. The receivers change addresses whenever they successfully copy a new message. Receivers that miss a message stay with the old address and eventually receive a retransmission; receivers that copy a message are spared receiving any future retransmissions. The parity system requires restrictions on the packet lifetime and outstanding messages that are rarely a problem in a local network. The ultimate in performance is achieved by a network interface that performs duplicate detection, ACK generation, and ACK reception without host intervention. These interfaces are referred to as *filter* interfaces in further discussion. The parity scheme is a primitive example of automatic duplicate detection; various interfaces, such as the Hyperchannel[13] and others[14] incorporate automatic ACK generation, though at a low level in the protocol hierarchy. A scheme for a high-level version is described by Mockapetris[11]. These acknowledgments are transmitted immediately following the distribution they acknowledge, and hence are called *prompt* ACKs.**

These passages make no mention, teaching or suggestion of ...

> **"the outbound game payload enabling the gaming machine having sent the transaction packet to complete the game transaction and wherein the at least one gaming machine is configured such that a first arriving outbound payload received by the at least one gaming machine is effective to complete the game transaction, irrespective of when and if a second later arriving outbound payload is received by the at least one gaming machine."**

... as claimed in claim 1 and as similarly claimed in the other pending independent claims. Indeed, Mockapetris teaches for the interface to recognize multicast addresses instead of a single broadcast address and a *parity* scheme for duplicate **transmission** detection. So far, in Mockapetris, we are only talking about the initial transmission of the multicast messages, and have not yet reached the point where the ACKs are generated by the hosts (recipients of the multicast messages). Here, there is not teaching or suggestion therein for a sender to treat a first arriving outbound payload received by the at least one gaming machine as being effective to complete the game transaction, irrespective of when and if a second later arriving outbound payload is received by the at least one gaming machine, as claimed herein.

Mockapetris also wishes to optimize the acknowledgment algorithm (the generation and sending of the ACKs at the receiving end of the multi-cast transmission back to the original multicast message sender) and the associated burden that such acknowledgements pose at the recipients thereof (the original senders referred to above):



> We also want to optimize the acknowledgment algorithm. In multicast, there is more distinct acknowledgment information than data to be acknowledged; hence special acknowledgment algorithms may be justified.

Mockapetris does this through one of four multicast algorithms:

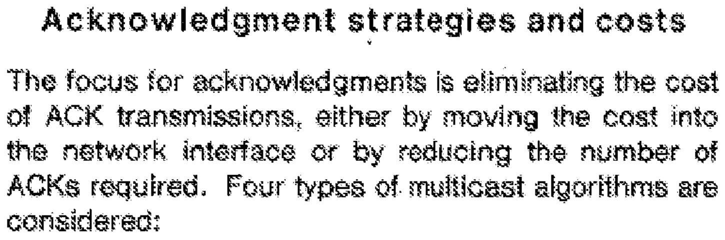
1. Simulation algorithms;

2. Multiple acknowledgment algorithms;

3. Saturation algorithms, and

4. Negative acknowledgment algorithms.

These algorithms are concerned with the **cost of sending of acknowledgments** by the hosts to the sender of the original multicast message.



Acknowledgment strategies and costs

The focus for acknowledgments is eliminating the cost of ACK transmissions, either by moving the cost into the network interface or by reducing the number of ACKs required. Four types of multicast algorithms are considered:

Even when Mockapetris states that "there is more distinct acknowledgement information than data to be acknowledged", he does not teach that a first arriving outbound payload received by the at least one gaming machine is treated as being effective to complete the game transaction, irrespective of when and if a second later arriving outbound payload is received by the at least one gaming machine, as claimed herein. This is simply because such a scheme would be contrary to the purpose of multicasting messages, as defined by Mockapetris: to insure that all intended recipients of the message receive the message. In contrast, the claimed embodiments only require a single outbound payload to be returned (irrespective of how many transaction packets were sent to the central servers) to complete the game transaction. How many others are subsequently received (if any), does not affect the completion of the transaction, according to the claimed embodiments.

Contrary to Mockapetris, the claimed embodiments are, again, unconcerned with the costs of ACK transmissions by either moving the burden to a network interface of by reducing the number of ACKs required.

Indeed, in the claimed embodiments, irrespective of the number of ACKS generated by the central servers, **a first arriving outbound payload received by the at least one gaming machine is treated as being effective to complete the game transaction, irrespective of when and if a second later arriving outbound payload is received by the at least one gaming machine**, as claimed herein. Mockapetris simply does not teach or suggest that the sender of the multicast message, after having sent same, discards all but the first-to-arrive ACK received from the ACK-sending hosts.

At no point is Mockapetris believed to teach or suggest the utility or desirability of discarding ACKs, after a first in time ACK has been received by the sender of the multicast. The whole point of multicasting is the transmission of a message to predetermined multiple receivers and to ensure that the intended receivers thereof actually received the multicast messages:

In the **Simulation Algorithm** (number 1 above), each host sends an ACK back to the sender or a software ring is used, in which the last destination acknowledges receipt to the sender.

The **Separate Acknowledgment Algorithm** (number 2 above) relies on one-to-one ACKs, which can result in the cost of acknowledgements being greater than the cost of the message distribution (again, here we are talking about the cost of generating and sending the ACKs, of which the claimed embodiments are agnostic).

The **Saturation Algorithm** (number 3 above), does not even rely on ACKs, but on the transmission of a sufficient number of copies of the message to insure that at least one copy thereof gets through to each destination.

Lastly, the **Negative Acknowledgment Algorithm** (number 4 above) in which members of the multicast that receive the message correctly do not send an ACK and those members of the multicast that would like to be able to copy the message, but cannot, send a NACK.

Each of these algorithms is solely concerned with the manner in which the host generates and sends the ACKs (or not) back to the multicast message sender. The claims, on the other hand, are crafted from the point of view of the receiver of the ACKs (to use Mockapetris' terminology) – that is, from the point of view of the claimed gaming machine, not from the point of view of the central servers that generate and send the outbound payloads (which the Exr. has analogized to Mockapetris' ACKs). None of these algorithms teach or suggest, whether considered alone or in combination with Nguyen (which teaches gaming machines), treating a first arriving outbound payload received by the at least one gaming machine as being effective to complete the game transaction, irrespective of when and if a second later arriving outbound payload is received by the at least one gaming machine, as claimed herein.

It is believed, therefore, that the independent claims define an online gaming systems and methods that find no counterpart in the applied Mockapetris-Nguyen combination.

Applicants' attorney believes that the present application is now in condition for allowance and passage to issue. If any unresolved issues remain, the Examiner is respectfully invited to contact the undersigned attorney of record at the telephone number indicated below, and whatever is required will be done at once.

Respectfully submitted,

Date: January 31, 2011 By:

Alan W. Young
Attorney for Applicants
Registration No. 37,970

YOUNG LAW FIRM, P.C.
4370 Alpine Rd., Ste. 106
Portola Valley, CA 94028
Tel.: (650) 851-7210
Fax: (650) 851-7232

C:\YLF\CLIENTS\CYBS\5872 (Univ Gaming Server)\5872 Proposed Amendments in preparation for Tel interview of 2-1-2011.doc